

Bluestock Fintech

Technical Documentation: IPO Web App & REST API

Bluestock Fintech

1. INTRODUCTION

The Bluestock IPO Web App & REST API is a full-stack application designed to manage and present Initial Public Offering (IPO) data to users. It supports user registration, login, IPO listings, detail views, and applications. This project was built during an internship at Bluestock Fintech.

2. FRONTEND

2.1 Technologies

- Language: JavaScript
- Framework: React (CRA)
- Libraries:
 - Axios (HTTP requests)
 - Tailwind CSS (Styling)
 - React Router DOM (Navigation)
- Version Control: Git (private repo)

2.2 Folder Structure

```
frontend/  
├── public/  
├── src/  
|   ├── components/
```

```
| /— pages/  
| /— layouts/  
| /— App.js  
| /— index.js  
  
/— package.json
```

2.3 Setup Instructions

```
cd frontend
```

```
npm install
```

```
npm start
```

Access at: `http://localhost:3000`

3. BACKEND

3.1 Technologies

- Language: JavaScript
- Framework: Node.js + Express
- Libraries:
 - Mongoose (MongoDB interaction)
 - BcryptJS (Password hashing)
 - dotenv (Environment variables)
 - cors (Cross-Origin Resource Sharing)
- Version Control: Git (private)

3.2 Folder Structure

```
backend/
```

```
└─ models/
  └─ User.js

└─ controllers/
  └─ authController.js

└─ routes/
  └─ authRoutes.js

└─ app.js

└─ .env.example

└─ package.json
```

3.3 Setup Instructions

```
cd backend

npm install

cp .env.example .env

node app.js
```

Access API at: `http://localhost:5000`

4. DATABASE

Primary: MongoDB (Cloud - MongoDB Atlas)

Schema:

- Users: `{ _id, name, email, password }`

- IPOs: `{ _id, company, price, openDate, closeDate }`
- Applications (planned): `{ _id, userId, ipoId, quantity }`

5. API ENDPOINTS

Endpoint	Method	Description	Parameters	Response
----- ----- ----- ----- -----				
/api/auth/register	POST	Register new user	name, email, password	JSON message
/api/auth/login	POST	User login	email, password	JSON message / token
/api/ipos	GET	Get all IPOs	None	List of IPOs
/api/ipos/:id	GET	IPO detail by ID	URL param `id`	IPO object
/api/applications	POST	Apply for IPO (planned)	ipoId, quantity	Application response

6. ARCHITECTURE

- Frontend: Single Page Application (React)
- Backend: REST API using Express (MVC style)
- Data Flow: React (client) → Express API → MongoDB Atlas → API → Client
- Components:
 - Reusable frontend components (Navbar, Layout)
 - Modular backend controllers, routes, models

7. TESTING

- Manual testing via:
 - Postman (API)
 - Browser (React UI)
- Unit tests (future work) planned with Jest and Supertest

8. DEPLOYMENT

 *Per Bluestock IP policy, no public deployment allowed.*

- Platform: Local development only
- Steps:
 1. Clone frontend and backend repos
 2. Setup `.env`` and install dependencies
 3. Run backend with ``node app.js``
 4. Run frontend with ``npm start``

9. SECURITY

- Passwords hashed using ``bcryptjs``
- Input validated at controller level
- Environment variables managed using `.env``
- CORS configured to allow frontend origin only
- Authentication middleware planned (JWT)

10. PERFORMANCE

- Frontend: Optimized with Tailwind, SPA routing
- Backend: Clean routes/controllers separation
- Database: MongoDB indexing planned

11. VERSION CONTROL

- Git + GitHub (private repo)
- Branching:
 - `main`: stable
 - `feature/`: feature-specific branches

12. FUTURE IMPROVEMENTS

- Add JWT-based login and user sessions
- Admin dashboard for IPO creation
- IPO application & status tracking
- Real-time data updates (WebSocket)
- Unit + Integration testing
- Dockerization for deployment